

# Módulo 2: Ejercicio de evaluación final

- Antes de empezar, hay que crear un nuevo repositorio desde GitHub Classroom usando [este enlace](#). Una vez creado, hay que clonar en nuestro ordenador y en la carpeta creada empezaremos a trabajar en el ejercicio.
- A continuación, **si quieres**, hay que descargar e incluir en el proyecto el [starter kit de Adalab](#).

## Enunciado

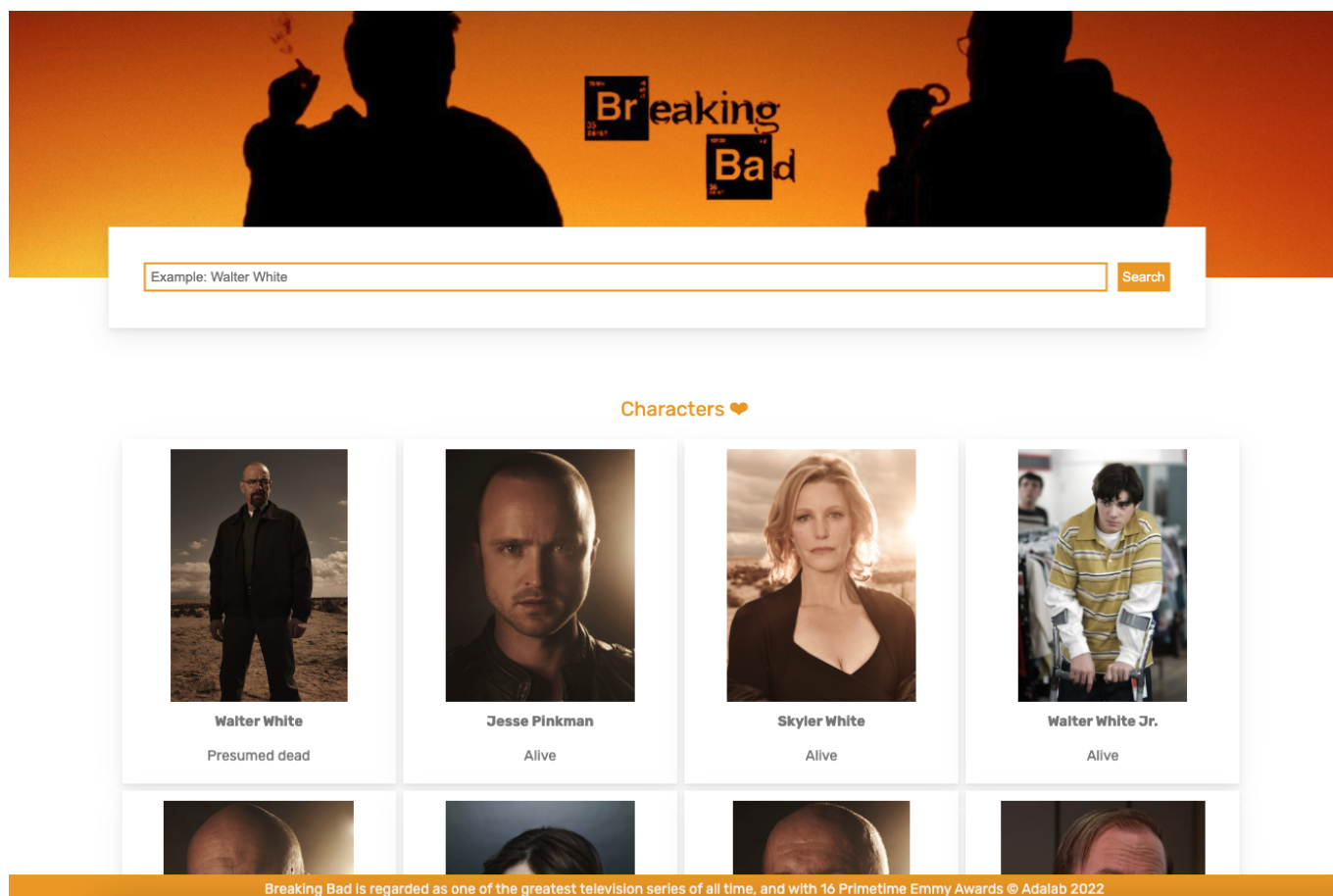
El ejercicio consiste en desarrollar una aplicación web de Breaking Bad, que nos permite des/marcar los personajes como favoritas y guardarlas en local storage.

El ejercicio también tiene una parte de maquetación con HTML y Sass, os recomendamos dedicar esfuerzo a la maquetación una vez terminada la parte de JavaScript, ya que los criterios de evaluación están relacionados con esta última.

Vamos de definir los distintos hitos del ejercicio:

### 1. Estructura básica

En primer lugar hay que realizar una estructura básica sobre este modelo. No hay que preocuparse por las medidas, colores ni tipografía hasta un hito posterior.



La aplicación de búsqueda de serie consta de dos partes:

1. Un campo de texto y un botón para buscar personajes por su título.
2. Un listado donde aparece la foto del personaje, el nombre del personaje y si está vivo o muerto.

## 2. Inicio

- Al levantar la página debe mostrar todos los resultados de los personajes de la serie, la aplicación debe conectarse a [The Breaking Bad API](https://breakingbadapi.com/api/characters). Os recomendamos echar un vistazo al JSON que devuelve una petición de búsqueda de todos los personajes para ver qué datos son los que necesitamos: <https://breakingbadapi.com/api/characters>.
- Por cada personaje obtenido en el resultado de la búsqueda hay que pintar una tarjeta donde mostramos la foto del personaje, el nombre del personaje y si está vivo o muerto.
- Para pintar la información en la página se puede elegir entre hacerlo de forma básica con `innerHTML` o manipulando de forma avanzada el DOM.

## 3. Búsqueda

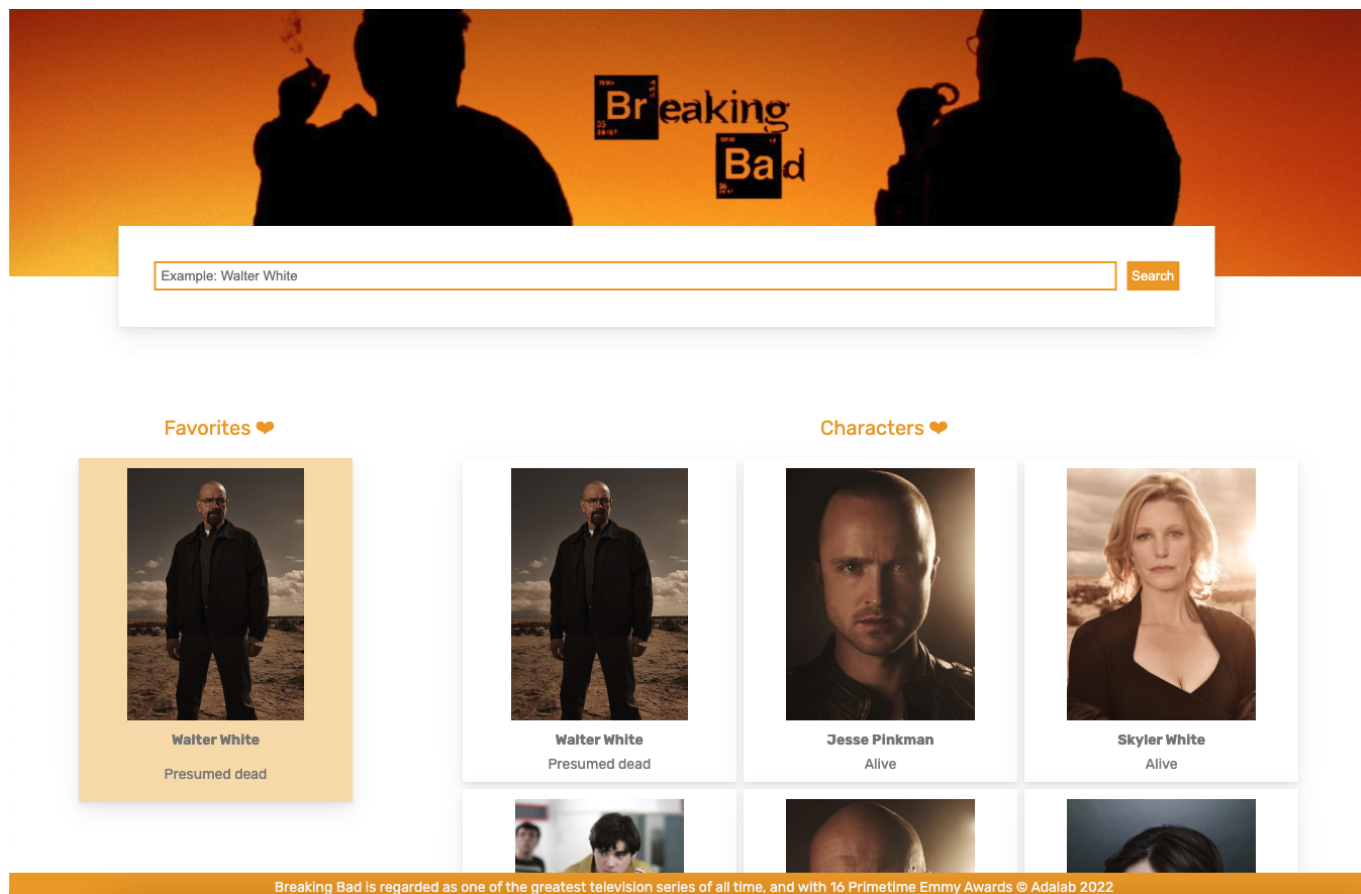
La usuaria tiene la opción de buscar personajes por su nombre, al hacer clic sobre el botón de **Buscar**, la aplicación solo debe mostrar aquellos personajes que coinciden con la búsqueda realizada por la usuaria:

- Puedes hacer un filtro por nombre sobre el listado de personajes obtenido anteriormente.
- Puedes utilizarla siguiente url con el parámetro `name` para obtener del API solo los usuarios que coincidan con el nombre especificado por la usuaria: <https://breakingbadapi.com/api/characters?name=Walter>

## 4. Favoritos

La usuaria puede indicar cuáles son sus personajes favoritos. Para ello, al hacer clic sobre un personaje debe pasar lo siguiente:

- Cambiar los estilos de la tarjeta del personaje para indicar que es un personaje favorito.
- Mostrar un listado con los personajes favoritos. Os recomendamos crear un variable o constante de tipo array en JS para almacenar los personajes favoritos.
- Los personajes favoritos deben seguir apareciendo en la página aunque la usuaria realice otra búsqueda.



## 5. Almacenamiento local

Hay que almacenar el listado de favoritos en el localStorage. De esta forma, al recargar la página el listado de favoritos debe mostrarse.

## 6. BONUS: Borrar favoritos

Como bonus, os proponemos la opción de borrar favoritos. Al hacer clic sobre el icono de una 'x' al lado de cada favorito, hay que borrar el favorito clicado de la lista y del localStorage.

Para terminar de rematar nuestra aplicación de BreakingBad, nos gustaría poder añadir/quitar como favorito al hacer click sobre un personaje. Y que, si realizamos una nueva búsqueda y sale un personaje que ya es favorito, aparezca resaltado en los resultados de búsqueda (con algún estilo de CSS).

Y ya sería fantástico si al final de la lista de favoritos hay un botón para borrar todos los favoritos a la vez.

## 7. BONUS: Afinar la maquetación

Una vez terminada la parte de interacción, podemos centrarnos en la parte de maquetación donde tenéis libertad para decidir los estilos.

## Entrega

El límite de entrega es el martes 15 de noviembre a las 14:00.

Solo debéis hacer commits y merges en la rama master de vuestro repositorio hasta la fecha límite. Si después del ejercicio queréis seguir trabajando sobre el ejercicio, lo podéis hacer en otra rama y no debéis

combinarla (merge) hasta que los profesores os lo indiquen.

La evaluación solo se considerará terminada cuando:

- Esté publicada en GitHub Pages y esté funcionando.
- El enlace a GitHub Pages esté en la página principal del repositorio, en la parte superior, al lado de la descripción.

## Normas

Este ejercicio está pensado para que lo realices de forma individual en clase, pero podrás consultar tus dudas con la profesora y tus compañeras si lo consideras necesario. Ellas no te darán directamente la solución de tu duda, pero sí pistas para poder solucionarla. Aún facilitando la comunicación entre compañeras, durante la prueba no debes copiar código de otra persona ni acceder a su portátil. Confiamos en tu responsabilidad.

La evaluación es una buena oportunidad para conocer cómo estás progresando, saber qué temas debes reforzar durante las siguientes semanas y cuáles dominas. Te recomendamos que te sientas cómoda con el ejercicio que entregues y no envíes cosas copiadas que no entiendas.

Si detectamos que has entregado código que no es tuyo, no entiendes y no lo puedes defender, pasarás directamente a la re-evaluación del módulo. Tu objetivo no debería ser pasar la evaluación sino convertirte en programadora, y esto debes tenerlo claro en todo momento.

Una vez entregado el ejercicio realizarás una revisión del mismo con la profesora (25 minutos), que se asemejará a una entrevista técnica: te pedirá que expliques las decisiones tomadas para realizarlo y te propondrá realizar cambios in situ sobre tu solución.

Es una oportunidad para practicar la dinámica de una entrevista técnica donde te van a proponer cambios sobre tu código que no conoces a priori. Si evitas que otras compañeras te den pistas sobre la dinámica de feedback, podrás aprovecharlo como una práctica y pasar los nervios con la profesora en lugar de en tu primera entrevista de trabajo.

Al final tendrás un feedback sobre aspectos a destacar y a mejorar en tu ejercicio, y sabrás qué objetivos de aprendizaje has superado de los listados a continuación.

## Criterios de evaluación

Vamos a listar los criterios de evaluación de este ejercicio. Si no superas al menos el 80% de estos criterios o no has superado algún criterio clave (marcados con \*) te pediremos que realices una re-evaluación con el fin de que termines el curso mejor preparada y enfrentes tu primera experiencia profesional con más seguridad. En caso contrario, estás aprendiendo al ritmo que hemos pautado para poder afrontar los conocimientos del siguiente módulo.

### General

- Usar una estructura adecuada de ficheros y carpetas para un proyecto web, y enlazar bien los distintos ficheros\*.
- Tener el código perfectamente indentado\*.
- Uso de control de versiones **con ramas** para manejar un proyecto de código.

## JavaScript básico

- Crear código JavaScript con sintaxis correcta y bien estructurado\*.
- Usar constantes / variables para almacenar información y re-assignar valores\*.
- Usar condicionales para ejecutar acciones distintas en función de una condición\*.
- Saber trabajar con listados de datos (arrays)\*.
- Usar funciones para estructurar el código.
- Saber modificar la información del DOM para añadir contenido dinámico\*.
- Saber escuchar eventos del DOM y actuar en consecuencia\*.

## Peticiones AJAX y APIs

- Crear peticiones con fetch y promesas\*.
- Saber trabajar correctamente con la respuesta del servidor\*.
- Gestionar información en formato JSON.
- Usar el localStorage para guardar información en el navegador.

## Issues

- Haber resuelto las issues de la evaluación intermedia.

## Otros criterios a tener en cuenta

- Usar inglés para nombres de variables, funciones, clases, mensajes de commit, nombres de ficheros.
- El repositorio de GitHub debe tener README explicando muy brevemente cómo arrancar el proyecto.

**¡Al turrón!** (o a la rica tortilla de patatas)